



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Název: VY_32_INOVACE_PG4119 Tipy a triky, rozdíly mezi prohlížeči

Autor: Mgr. Tomáš Javorský

Datum vytvoření: 06 / 2012

Ročník: 3

Vzdělávací oblast / téma: webdesign, počítačová grafika

Anotace: DUM seznamuje s nejčastějšími problémy, na které může webdesiner narazit, upozorňuje na některé rozdíly mezi prohlížeči a ukazuje, jak je řešit. Studenti se využitím tohoto materiálu naučí, jak dosáhnout shodného zobrazení stránek v různých prohlížečích nebo jak vytvořit dvousloupcové rozložení stránky s proměnnou výškou.

Tipy a triky, rozdíly mezi prohlížeči

Při vytváření webových stránek se musíme často vyrovnat s tím, že rozdílné prohlížeče interpretují různé vlastnosti odlišně, jinak přistupují k zadané velikosti prvků atd. Pro většinu situací existuje nějaké řešení, i když je třeba občas napsat několik řádek kódu navíc nebo vytvořit nějaký grafický prvek. Tento dokument obsahuje řešení některých nejčastějších problémů, s nimiž se lze při vytváření stránek setkat.

Rozdílné rozměry prvků v Internet Exploreru a Mozille

Jedním z problémů je to, že IE (zejména starší verze) vypočítává šířku a výšku objektů jinak než Mozilla a ostatní prohlížeče dodržující standardy - podívejme se na příklad:

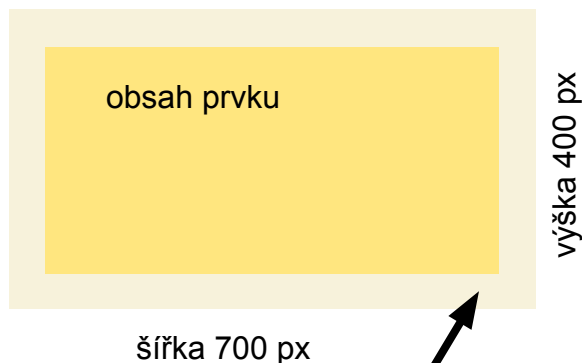
Máme prvek o šířce 700 px a výšce 400 px, s paddingem 50 px z každé strany.

Zápis v tabulce CSS:

```
div { width:700px; height:400px; padding:50px; }
```

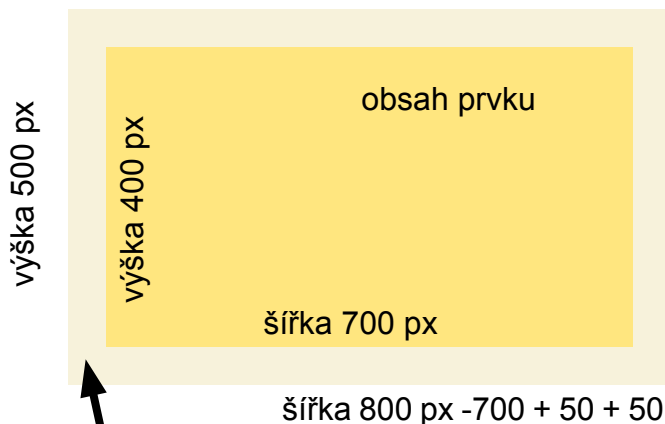
Zatímco IE ponechá vnější rozměry prvku a padding vytvoří dovnitř prvku, Mozilla vnější rozměry prvku zvětší o padding, takže je prvek větší - šířka: $700 + 50 + 50 = 800$ px, výška $400 + 50 + 50 = 500$ px. Totéž se týká i vlastnosti border - má-li prvek nastavený nějaký padding nebo border, zobrazí se jinak v IE a jinak v Mozille, což může vést k rozhození layoutu stránky.

Prvek v IE



padding se vytvoří dovnitř prvku, rozměry zadané v CSS zůstanou

Prvek v Mozille



padding se přidá k rozměrům prvku, rozměry zadané v CSS se zvětší o padding

Řešení:

Prvek, který má zadaný padding nebo border, ale nikoliv rozměry uzavřeme do vnějšího prvku (obalu), který má zadané rozměry, ale nemá zadaný padding ani border. Pokud nemá prvek padding ani border, zobrazí se stejně v obou prohlížečích, zůstanou zachovány jeho rozměry a proměnlivý bude pouze vnitřní prvek, takže nedojde k rozhození layoutu stránky - ukázka kódu na další straně.

Ukázka kódu:

```
#obal { width:500px; height:400px; }  
  
#vnitrek { padding:20px; border:2px solid red; }  
  
<div id="obal">  
<div id="vnitrek">  
</div>  
</div>
```

Jistou nevýhodou tohoto postupu je, že přidává další HTML prvky do kódu, nicméně protože prvky <div> nemají žádnou významovou hodnotu, nenarušují sémantiku kódu

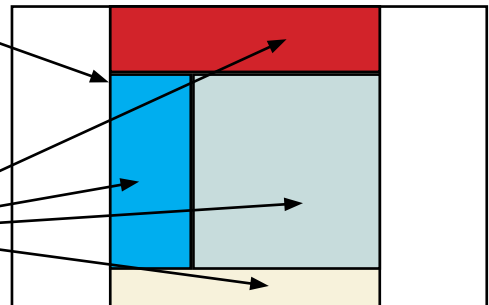
Vystředění layoutu v IE a v Mozille

Pokud máme rozložení stránky, které nepřesahuje celou obrazovku a potřebujeme umístit layout doprostřed obrazovky, musíme použít jiný přístup v Mozille a jiný v Exploreru.

Celý layout uzavřeme do prvku div, který má zadanou šířku - buď pevnou nebo procentuelní a pak tento div nastylujeme - ukázka následuje:

obalovy <div id="hlavniObal">, který uzavírá všechny prvky stránky

ostatní prvky stránky - menu, obsah atd.



```
<style>
```

```
body { text-align:center; }
```

Vystředí vše na stránce v IE. U dalších prvků je pak nutné nastavit znovu vlastnost text-align;

```
#hlavniObal { margin-left:auto; margin-right:auto; }
```

```
</style>
```

```
<body>
```

```
<div id="hlavniObal">
```

mezi tímto prvkem leží všechny ostatní prvky stránky

```
</div>
```

```
</body>
```

Nastavení stylu obalového divu: **margin-left:auto;** a **margin-right:auto;** vystředí div a všechno v něm v Mozille. V IE nebude mít vliv.

Absolutní reset

Většina prvků se v prohlížečích zobrazuje mírně odlišně - jde zejména o vlastnosti margin a padding, což se později špatně odlaďuje. Abychom zobrazení všech prvků v prohlížečích sjednotili, lze použít tzv. **absolutní reset**, kdy nastavíme všem prvkům nulový margin, padding a border, takže se budou zobrazovat ve všech prohlížečích stejně. Stačí do tabulky stylů zapsat následující definici - hvězdička na místě selektoru znamená, že se definice týká všech prvků v HTML dokumentu:

```
<style>
```

```
* { margin:0; padding:0; border:0; }
```

```
</style>
```

Falešný dvousloupcový layout vytvořený pomocí obrázku

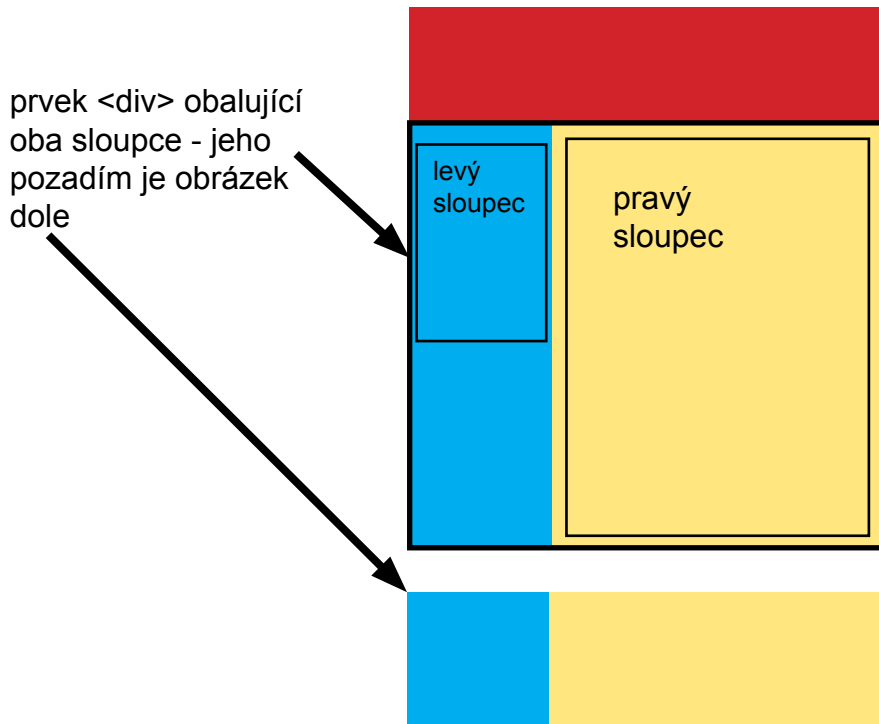
Pokud vytváříme dvousloupcový layout s různobarevnými sloupci, narážíme na problém, která část stránky bude vyšší - pokud bude vyšší pravá část, bude pod levým sloupcem vidět její barva (obr. 1) - a naopak. U layoutů, kde známe přesnou výšku sloupců, se tomu můžeme vyhnout, většinou to však není možné - obzvlášť u dynamicky generovaných webů - u těch může být levý i pravý sloupec delší. Existuje jedno elegantní a jednoduché řešení: Levý i pravý sloupec vytvoříme jako průhledný a prvku `<div>`, který je oba obaluje, přiřadíme obrázkové pozadí představující požadované dva sloupce. Pozadí se bude svisle opakovat, takže vyplní jakoukoliv výšku stránky.

obr. 1



protože je pravý sloupec delší,
bude pod levým sloupcem vidět
pozadí

obr. 2



obrázek (např. pozadi.jpg) tvořící pozadí
prvku a vytvářející efekt 2 sloupců

Ukázka kódu z předchozího příkladu:

```
<style>
#obaDvaSloupce { background-image:url("pozadi.jpg"); }

#levySloupec { float:left; width:150px; }

#pravysloupec { float:right; width:700px; }

</style>

<div id="obaDvaSloupce">

<div id="levySloupec">
obsah prvku
</div>

<div id="pravysloupec">
obsah prvku
</div>

</div>
```